

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ORCA SECURITY LTD,
Petitioner

v.

WIZ, INC.,
Patent Owner

Inter Partes Review Case No. IPR2025-01087
U.S. Patent No. 12,003,529

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 12,003,529**

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
II.	THE '529 PATENT.....	1
A.	SUMMARY OF THE '529 PATENT.....	1
B.	SUMMARY OF UNPATENTABILITY	3
C.	SUMMARY OF THE PROSECUTION HISTORY	4
D.	LEVEL OF SKILL OF A PERSON HAVING ORDINARY SKILL IN THE ART.....	5
III.	REQUIREMENTS UNDER 37 C.F.R. § 42.104	5
A.	GROUND FOR STANDING UNDER 37 C.F.R. § 42.104(A).....	5
B.	IDENTIFICATION OF CHALLENGE UNDER 37 C.F.R. § 42.104(B)	6
C.	CLAIM CONSTRUCTION UNDER 37 C.F.R. § 42.104(B)(3).....	6
IV.	SHOWING OF ANALOGOUS PRIOR ART	6
A.	SHUA	7
B.	LANG	8
V.	GROUND 1: CLAIMS 1-17 ARE OBVIOUS OVER SHUA IN VIEW OF LANG.....	10
A.	CLAIM 1	10
1.	<i>Claim 1[P]: “A method for detecting a cybersecurity risk of an artificial intelligence (AI)”</i>	<i>10</i>
2.	<i>Claim 1[a]: “generating an inspectable disk based on an original disk of a resource deployed in a computing environment”</i>	<i>13</i>
3.	<i>Claim 1[b]: “inspecting the inspectable disk for an AI model”</i>	<i>14</i>
4.	<i>Claim 1[c]: “generating a representation of the AI model in a security database, the security database including a representation of the computing environment”</i>	<i>17</i>
5.	<i>Claim 1[d]: “inspecting the AI model for a cybersecurity risk”</i>	<i>22</i>
6.	<i>Claim 1[e]: “generating a representation of the cybersecurity risk in the security database, the representation of the cybersecurity risk connected to the representation of the AI model in response to detecting the cybersecurity risk”</i>	<i>23</i>

7.	<i>Claim 1[f]: “initiating a mitigation action based on the cybersecurity risk”</i>	28
B.	CLAIM 2	30
1.	<i>Claim 2: “detecting an artifact of the AI model on the inspectable disk”</i>	30
C.	CLAIM 3	32
1.	<i>Claim 3: “inspecting the AI model to detect an AI model configured to execute a code object”</i>	32
D.	CLAIM 4	33
1.	<i>Claim 4: “detecting a metadata of the AI model, wherein the metadata indicates that the AI model is a cybersecurity risk”</i>	33
E.	CLAIM 5	36
1.	<i>Claim 5: “detecting in the AI model any one of: a secret, a certificate, a code, and any combination thereof”</i>	36
F.	CLAIM 6	39
1.	<i>Claim 6: “generating a lateral movement path, wherein the lateral movement path includes the AI model, and a representation of a resource, wherein the resource is accessible utilizing the AI model”</i>	39
G.	CLAIM 7	44
1.	<i>Claim 7: “applying a policy on the representation of the computing environment”</i>	44
H.	CLAIM 8	45
1.	<i>Claim 8: “initiating the mitigation action based on the policy”</i>	45
I.	CLAIM 9	46
1.	<i>Claim 9[P]: “non-transitory computer-readable medium storing a set of instructions for detecting a cybersecurity risk of an artificial intelligence (AI), the set of instructions comprising: one or more instructions that, when executed by one or more processors of a device, cause the device to”</i>	46
2.	<i>Claim 9[a]: “generate an inspectable disk based on an original disk of a resource deployed in a computing environment”</i>	48
3.	<i>Claim 9[b]: “inspect the inspectable disk for an AI model”</i>	48

4.	Claim 9[c]: “generate a representation of the AI model in a security database, the security database including a representation of the computing environment”	48
5.	Claim 9[d]: “inspect the AI model for a cybersecurity risk”	48
6.	Claim 9[e]: “generate a representation of the cybersecurity risk in the security database, the representation of the cyber-security risk connected to the representation of the AI model in response to detecting the cybersecurity risk”	48
7.	Claim 9[f]: “initiate a mitigation action based on the cybersecurity risk”	48
J.	CLAIM 10	49
1.	Claim 10[P]: “A system for detecting a cybersecurity risk of an artificial intelligence (AI) comprising: a processing circuitry; a memory, the memory containing instructions that, when executed by the processing circuitry, configure the system to ”	49
2.	Claim 10[a]: “generate an inspectable disk based on an original disk of a resource deployed in a computing environment”	50
3.	Claim 10[b]: “inspect the inspectable disk for an AI model”	50
4.	Claim 10[c]: “generate a representation of the AI model in a security database, the security database including a representation of the computing environment”	50
5.	Claim 10[d]: “inspect the AI model for a cybersecurity risk”	51
6.	Claim 10[e]: “generate a representation of the cybersecurity risk in the security database, the representation of the cyber-security risk connected to the representation of the AI model in response to detecting the cybersecurity risk”	51
7.	Claim 10[f]: “initiate a mitigation action based on the cybersecurity risk”	51
K.	CLAIM 11	51
1.	Claim 11: “detect an artifact of the AI model on the inspectable disk”	51
L.	CLAIM 12	51

1.	<i>Claim 12: “inspect the AI model to detect an AI model configured to execute a code object”</i>	51
M.	CLAIM 13	51
1.	<i>Claim 13: “detect a metadata of the AI model, wherein the metadata indicates that the AI model is a cybersecurity risk”</i>	51
N.	CLAIM 14	51
1.	<i>Claim 14: “detect in the AI model any one of: a secret, a certificate, a code, and any combination thereof”</i>	51
O.	CLAIM 15	52
1.	<i>Claim 15: “generate a lateral movement path, wherein the lateral movement path includes the AI model, and a representation of a resource, wherein the resource is accessible utilizing the AI model”</i>	52
P.	CLAIM 16	52
1.	<i>Claim 16: “apply a policy on the representation of the computing environment”</i>	52
Q.	CLAIM 17	52
1.	<i>Claim 17: “initiate the mitigation action based on the policy”</i>	52
VI.	CONCLUSION	52
VII.	MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(A)(1)	54
A.	REAL PARTY-IN-INTEREST	54
B.	RELATED MATTERS	54
C.	LEAD AND BACK-UP COUNSEL	54
D.	37 C.F.R. § 42.8(B)(4) – SERVICE INFORMATION	55

TABLE OF AUTHORITIES

Cases

<i>Elekta Ltd v. ZAP Surgical Sys., Inc.</i> , (Fed. Cir. 2023).....	13
<i>Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.</i> , (Fed. Cir. 2017)	6

Statutes

35 U.S.C. § 102(a)(2)	9
35 U.S.C. § 103	6
35 U.S.C. 102(a)(1)	7

Regulations

37 C.F.R. § 42.104.....	5
37 C.F.R. § 42.104(a)	5
37 C.F.R. § 42.104(b)	6
37 C.F.R. § 42.104(b)(3)	6
37 C.F.R. § 42.8(a)(1)	54
37 C.F.R. § 42.8(b)(1)	54
37 C.F.R. § 42.8(b)(3)	54

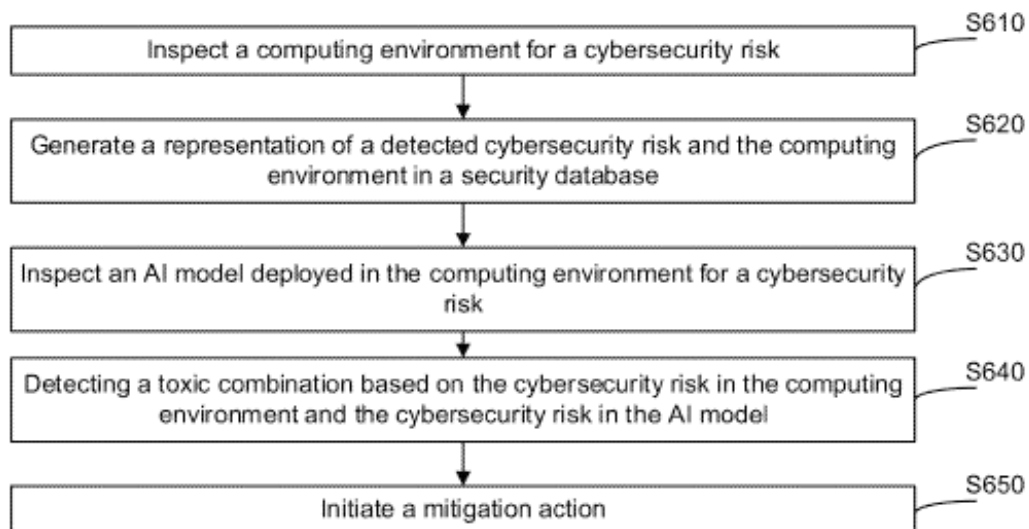
I. INTRODUCTION

Petitioner Orca Security Ltd. (“Petitioner”) requests *Inter Partes* Review of Claims 1–2 and 4–8 (the “Challenged Claims”) of U.S. Patent No. 12,003,529 (“the ’529 Patent”) assigned to Wiz, Inc. (“Patent Owner”).

II. THE ’529 PATENT

A. Summary of the ’529 Patent

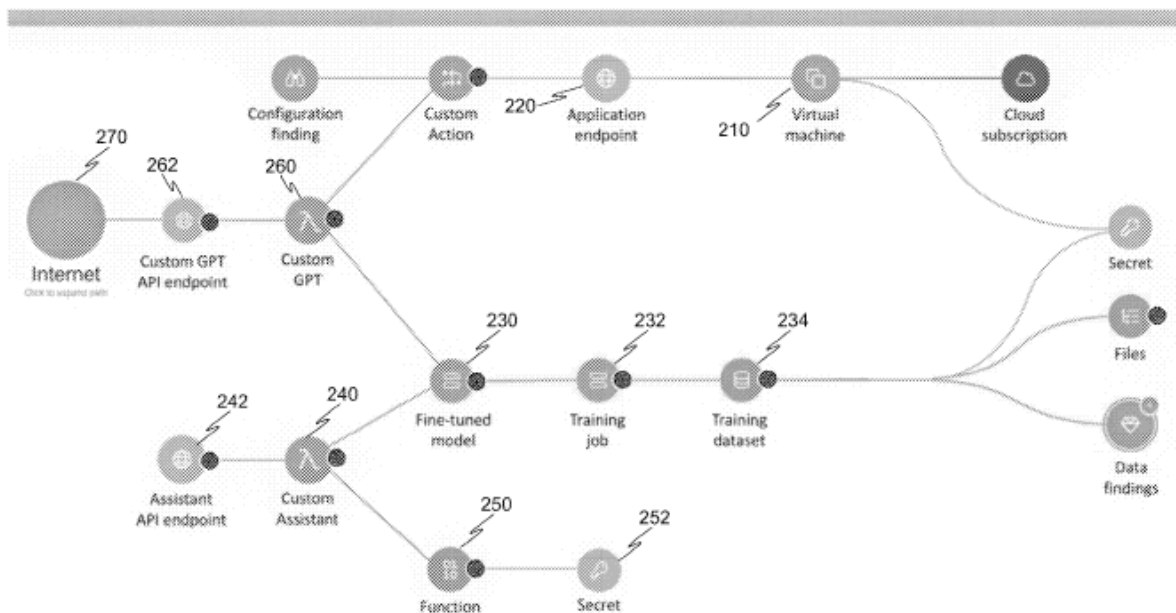
The ’529 Patent generally relates to detecting an artificial intelligence model cybersecurity risk in a cloud computing environment. *Ex. 1001*, ’529 Patent, 1:61-67. As illustrated in the ’529 Patent, this method begins with inspecting a cloud computing environment for a cybersecurity risk:



Id., FIG. 6. As laid out in the claims, this process includes substeps of generating an inspectable disk based on an original disk (by, for example, creating a snapshot of

the original disk), and inspecting the inspectable disk (e.g., snapshot) for an AI model. *Id.* 15:45-51, 15:32-36.

Next, the process of the '529 Patent generates a “representation” of the detected AI model and the “computing environment” in a “security database” (e.g. a graph of the cloud computing environment). *Id.*, 7:46-60. A representative illustration of this “security database” is depicted in the '529 Patent:



Id., FIG. 2. As characterized by the '529 Patent, this security database can be a graph database comprising nodes and connections between them. *Id.* 8:34-39.

The process of the '529 Patent continues by inspecting the AI model for a cybersecurity risk. *Id.*, 16:9-15. As examples of a “cybersecurity risk” in the AI model, the '529 Patent gives processes such as “determining if the AI model matches a version, identifier, and the like, of an AI model which is previously determined as

a cybersecurity risk” or “detecting a secret, a code, and the like, ... which the AI model, when prompted with an appropriate prompt, is configured to output, leading to leaked data, leaked secrets, etc.” *Id.*, 16:21-24; 16:28-31.

Once a “cybersecurity risk” in the AI model is identified, a “finding” (e.g., a data recording indicating the detected vulnerability) is added to the security database in connection with the AI model. *Id.*, 15:9-15. The process of the ’529 Patent concludes by initiating a mitigation action for the cybersecurity risk. The ’529 Patent characterizes these mitigations broadly, giving examples including “revoking access from a principal, revoking access to a resource, revoking access from a resource, sandboxing a resource, revoking access to an AI model, revoking access from an AI model, denying network communication directed to the AI model, such as network communication including a prompt for the AI model, generating an alert, updating an alert, generating an alert severity, [and] updating an alert severity.” *Id.*, 17:25-31.

B. Summary of Unpatentability

The ’529 Patent broadly describes a minor modification of the well-known and widely practiced cloud cybersecurity technique of scanning a customer’s cloud footprint to generate a graph of the interconnections between the cloud assets and identify known vulnerabilities that might allow unauthorized access to sensitive data. *See generally* Ex. *Shua* 1004. The purported advancement of the ’529 Patent is scanning for and identifying vulnerabilities in AI models, in addition to doing so for

other cloud assets. Ex. 1001, '529 Patent, 1:15-35. However, locating AI models and identifying security vulnerabilities in AI models was also well known before the filing date of the '529 Patent. *See* Ex. 1006, *Lang*, [0030] (locating AI models in virtual machine environments); [0021] (identifying vulnerabilities in AI models). Particularly, given the emphasis in cloud scanning systems on enumerating and mapping all of customers' cloud assets (*see* Ex. 1005, *Shua Provisional*, [0014], [0029], [0069]), it would have been obvious to modify an existing cloud scanning system to scan for AI models and identify vulnerabilities therein. Ex. 1003, ¶ 60.

C. Summary of the Prosecution History

U.S. Patent Application 18/584659, from which the '529 Patent issued, was filed on Feb. 22, 2024. Ex. 1002, '529 *File History*, p. 154. A first Office Action was issued on Apr. 10, 2024, rejecting the independent claims (as-filed claims 1, 10, and 11) as obvious under 35 U.S.C § 103 over U.S. Patent Application Publication US 2023/0269272 A1 to Dambrot et al., in view of U.S. Patent Application Publication US 2023/0269272 A1 et al. *Id.*, pp. 76-77. Other claims (including as-filed claims 2 and 12) were indicated as overcoming the prior art and being allowable if incorporated into the independent claims. *Id.*, p. 75.

Two days later, on April 12, 2024, Applicant files a response amending the independent claims to recite steps of “generating an inspectable disk based on an original disk of a resource deployed in the computing environment; and inspecting

the inspectable disk for the AI model,” material initially recited in as-filed claims 2 and 12. *Id.*, pp. 65-71. Less than two weeks later, on April 24, 2024, the Examiner issued a Notice of Allowance citing the amendments as the reason for allowance, and the ‘529 Patent issued on Jun. 4, 2024. *Id.* pp. 14-19; 1-2.

However, as the discussion below shows, the limitation of “generating an inspectable disk based on an original disk of a resource deployed in the computing environment” was well-known in the art as of the filing date of the ‘529 Patent, and “inspecting an inspectable disk for an AI model” was similarly obvious. *See* limitations 1[a], 1[b] *infra*.

D. Level of Skill of a Person Having Ordinary Skill in the Art

A POSITA at the time of the ‘529 Patent would have had a bachelor’s degree in computer engineering, computer science, electrical engineering, or a similar degree, with 2-3 years of experience in the fields of cybersecurity and artificial intelligence; additional education or experience might substitute for these requirements. Ex. 1003, ¶¶ 32-33.

III. REQUIREMENTS UNDER 37 C.F.R. § 42.104

A. Grounds for Standing Under 37 C.F.R. § 42.104(a)

Petitioner certifies the ‘529 Patent is available for IPR and Petitioner is not barred or otherwise estopped from requesting IPR as set forth in this Petition.

B. Identification of Challenge Under 37 C.F.R. § 42.104(b)

Petitioner requests the Challenged Claims be found unpatentable on the following grounds.

Proposed Grounds of Unpatentability	Exhibits
Ground 1: Claims 1–17 are obvious under 35 U.S.C. § 103 ¹ over Shua in view of Lang	1004 1005 1006

C. Claim Construction Under 37 C.F.R. § 42.104(b)(3)

In the district court litigation, the parties have not identified any claims as requiring construction. *See* Ex. 1021, *Joint Claim Construction Statement*. Accordingly, Petitioner submits that no claim terms require construction to determine the validity of the challenged claims under the Proposed Grounds of Unpatentability above. *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (construction is required “only to the extent necessary to resolve the controversy”).

IV. SHOWING OF ANALOGOUS PRIOR ART

The ’529 Patent was filed on Feb. 22, 2024, and does not claim priority to any earlier applications. Accordingly, the earliest possible priority date for the ’529 Patent is Feb. 22, 2024.

¹ All references to Title 35 are to post-AIA Title 35 unless otherwise stated.

A reference is analogous art to the claimed invention if: (1) the reference is from the same field of endeavor as the claimed invention (even if it addresses a different problem); or (2) the reference is reasonably pertinent to the problem faced by the inventor (even if it is not in the same field of endeavor as the claimed invention). MPEP 2141.01(a)(I).

A. Shua

U.S. Patent No. 11,489,863 to Shua (Ex. 1004, “Shua”) is prior art to the ’529 Patent. Shua was filed on Apr. 8, 2022, claiming priority to provisional application 63/180,048 (Ex. 1005, “*Shua Provisional*”), filed on Apr. 26, 2021. *See* Ex. 1004, *Shua*. Shua issued on Nov. 1, 2022. *Id.* Shua therefore qualifies as prior art to the ’529 Patent under 35 U.S.C. §§ 102(a)(1) and (2). Shua was not cited or considered during the prosecution of the ’529 Patent. *See* Ex. 1002, ’529 File History.

Shua incorporates the disclosures of the Shua Provisional by reference. Ex. 1001, *Shua*, 1:6-10. Accordingly, the disclosures of the Shua Provisional are “as much a part of the application as filed as if the text was repeated in the application, and should be treated as part of the text of the application as filed.” MPEP 2163.07(b). As such, this Petition relies on the disclosures of Shua and the Shua Provisional as a single prior art reference.

Shua is analogous prior art to the ’529 Patent because Shua is in the same field of endeavor as the ’529 Patent and is reasonably pertinent to a problem with which

the inventors of the '529 Patent were involved. Ex. 1003, ¶¶ 65-66. First, like the '529 Patent, Shua is directed to the field of endeavor of security in cloud computing environments. *Cf.* Ex. 1001, '529 Patent, 1:8-11 (“The present disclosure relates ... specifically to implementing cybersecurity monitoring solutions on an AI development pipeline of a cloud computing environment.”) *with* Ex. 1004, Shua, 1:20-23 (“Disclosed embodiments include ... securing cloud infrastructure”); *see also* Ex. 1003, ¶ 65. Second, Shua is reasonably pertinent to the problem being solved by the inventors of the '529 Patent: “that of implementing cybersecurity monitoring solutions on ... a cloud computing environment.” Ex. 1001, '529 Patent, 1:8-11. In particular, Shua “provide[s] technical solutions to challenges associated with cloud cybersecurity.” Ex. 1004, Shua, 6:33-34; *see also* Ex. 1003, ¶ 66. Accordingly, Shua satisfies both the “same field of endeavor” and “reasonably pertinent” prongs and is analogous prior art to the '529 Patent.

B. Lang

U.S. Patent Application No. US 2024/0202405A1 to Lang et al., published on Jun. 20, 2024, was the pre-grant publication of application 18/196,950, which was filed on May 12, 2023 and claims priority to provisional application 63/342,049, filed on May 13, 2022, provisional application 63/396,287, filed on Aug. 9, 2022, provisional application 63/443,859, filed on Feb. 7, 2023, and provisional application 63/458,741, filed on Apr. 12, 2023. Accordingly, the latest possible

effective filing date for Lang is May 12, 2023, which antedates the earliest possible priority date for the '529 Patent of Feb. 22, 2024. As such, Lang is prior art to the '529 Patent under at least 35 U.S.C. § 102(a)(2). Lang was not cited or considered during the prosecution of the '529 Patent. *See* Ex. 1002, '529 File History.

Lang is analogous prior art to the '529 Patent because Lang is in the same field of endeavor as the '529 Patent and is reasonably pertinent to a problem with which the inventors of the '529 Patent were involved. Ex. 1003, ¶¶ 70-74. First, like the '529 Patent, Lang is directed to the field of endeavor of security in cloud computing environments. *Cf.* Ex. 1001, '529 Patent, 1:8-11 (“The present disclosure relates generally to detecting artificial intelligence models, and specifically to implementing cybersecurity monitoring solutions on an AI development pipeline of a cloud computing environment.”) *with* Ex. 1006, *Lang*, [0003] (“**embodiments of the invention are directed at automatically and semi-automatically training, detecting, analyzing, monitoring, protecting, and hardening AI systems** and their components, including dependencies, binaries, source code, data, etc., to **improve the system’s** trust, reliability, accuracy, fairness, **security, etc., and protect it from potential vulnerabilities**, weaknesses, threats, limitations, etc.”); *see also* Ex. 1003, ¶¶ 70-71. Second, Lang is reasonably pertinent to the problem being solved by the inventors of the '529 Patent: “that of implementing cybersecurity monitoring solutions on an AI development pipeline of a cloud computing environment.” Ex.

1001, '529 Patent, 1:8-11. In particular, Lang states that “[a]nalysis may occur at any stage of the **AI development pipeline** ... For example, the analysis action may work alongside other actions. Analyses might pair with hardening, wherein **the model may be automatically hardened to protect it from vulnerabilities discovered during analysis.**” Ex. 1006, *Lang*, [0121]; *see also id.* [0030]; Ex. 1003, ¶¶ 72-73. Accordingly, Lang satisfies both the “same field of endeavor” and “reasonably pertinent” prongs and is analogous prior art to the '529 Patent.

V. GROUND 1: CLAIMS 1-17 ARE OBVIOUS OVER SHUA IN VIEW OF LANG

A. Claim 1

1. *Claim 1[P]: “A method for detecting a cybersecurity risk of an artificial intelligence (AI)”*

To the extent the preamble is limiting, Shua in view of Lang renders obvious *a method for detecting a cybersecurity risk of an artificial intelligence*² as claimed. In particular, Shua discloses techniques for detecting risk in a cloud environment (i.e., a method for detecting a cybersecurity risk): “**Embodiments of the present disclosure may deliver**³ instant-on, workload level **visibility across cloud infrastructure assets** ... helping organizations **to: [d]etect risks** such as vulnerabilities, malware, misconfigurations, lateral movement risk, and unsecured

² Requester denotes claim language by *italics*.

³ **Bold emphasis** is added throughout.

sensitive data[.]” Ex. 1004, Shua, 2:66-3:7. Shua discloses techniques for detecting a cybersecurity risk across all of an organization’s assets in a cloud environment, but does not expressly disclose detecting a cybersecurity risk of *an artificial intelligence*.

However, Lang discloses “a method for analyzing a computing system for properties of a machine learning model” in order “to ensure AI systems are working as intended,” because they are “difficult to audit.” Ex. 1006, Lang, *Abstract*; [0006]; *see also* [0020] (it is an object of Lang’s invention to “determine ... vulnerabilities” in AI computing systems). For example, Lang teaches that AI systems are “prone to adversarial attacks, wherein an attack manipulates the AI model or data (incl. training or inference data) in a way that affects the model’s behavior and decision-making.” *Id.* A POSITA would understand that Lang’s machine learning models (also referred to as AI systems) would have been a part of an organization’s assets. Ex. 1003, ¶¶ 82-84.

A POSITA would have been motivated to incorporate Lang’s scanning of AI models into Shua’s techniques for detecting risk in a cloud environment with a reasonable expectation of success for multiple reasons. First, a POSITA would have found it obvious to incorporate Lang’s teachings of scanning an AI model into Shua’s system for detecting risks across all of an organization’s assets, at least because Shua provides an express motivation to do so by teaching that it is desirable to scan “100% of an organization’s assets” Ex 1005, *Shua Provisional*, [0029]; Ex.

1003, ¶ 76. A POSITA would further understand, from Shua’s teachings (that it is desirable to scan 100% of an organization’s assets to detect risks) and Lang’s teachings (that AI systems are prone to adversarial attacks) that it would have been desirable to scan Lang’s artificial intelligence models to detect any cybersecurity risks present therein. Ex. 1006, Lang, [0006]; *id.* ¶ 84. A POSITA would have known that AI models exist as an assortment of possible assets (e.g., data/model files, code and scripts, configurations, and more). *Id.*

Second, such a combination would have constituted the use of a known technique (Lang’s scanning of AI models) to improve a similar device (Shua’s system for scanning an organizations assets) in the same way (Shua’s system would be improved in the same way as Lang’s, because the additional cybersecurity risks present in the AI models would be detected). Ex. 1003, ¶ 79. Additionally, both references describe scanning resources to detect cybersecurity risks. *Cf.* Ex. 1005, Shua Provisional, p. 57 (“A system for generating a composite cybersecurity risk assessment in a cloud environment”) *with* Ex. 1006, Lang, at [0274] (“Before, during and/or after deployment, it may check the system for cybersecurity risks”). Incorporating Lang’s scanning of AI models into Shua’s system for scanning all of an organization’s assets would have required at most minor modifications in software. Ex. 1003, ¶ 79. For at least these same reasons, a POSITA would have enjoyed a reasonable expectation of success in making such a combination. *Elekta*

Ltd v. ZAP Surgical Sys., Inc., 81 F.4th 1368, 1377 (Fed. Cir. 2023) (“[E]vidence establishing a motivation to combine may establish a finding of reasonable expectation of success.”)

2. ***Claim 1[a]: “generating an inspectable disk based on an original disk of a resource deployed in a computing environment”***

Shua teaches *generating an inspectable disk based on an original disk of a resource deployed in a computing environment*. As an initial matter, the ’529 specification gives creating a “snapshot of the original disk” as one example of creating “an inspectable disk based on an original disk of a resource.” Ex. 1001, ’529 *Patent*, 15:47-51 (“In some embodiments, **generating an inspectable disk includes generating** a clone of an original disk, **a snapshot of an original disk**, a copy of an original disk, a combination thereof, and the like, and inspecting the cloned disk for a cybersecurity object.”). The ’529 Patent further gives a “virtual machine” as an example of a “resource.” Ex. 1001, ’529 *Patent*, 4:64-67. (“**In some embodiments, a resource is a virtual machine**, a software container, a serverless function, an application, a database, a combination thereof, and the like.”). Finally, the ’529 Patent gives a “cloud computing environment” as one example of a “computing environment,” and “Amazon® Web Service (AWS)” as one example of a “cloud computing environment.” *Id.*, 11:53-56; 4:49-52. Thus, in the terms of the ’529 Patent, generating a snapshot of an original disk of a virtual machine deployed in

AWS is generating an inspectable disk based on an original disk of a resource deployed in a computing environment.

Shua expressly describes generating a snapshot of disk state of a machine deployed in a customer's AWS deployment (e.g., volumes within a cloud computing environment):

A snapshot of the disk state may be created through AWS or other cloud providers while the machine is running. The process can snapshot multiple volumes at the same point in time, resulting in deep and consistent visibility.

Ex. 1005, *Shua Provisional*, [0041]; *see also* [0039] (“For each region (e.g. geographic location) in a customer's cloud account, **Orca may enumerate all the possible compute assets, taking snapshots of their data volumes** to share with the Orca production account.”); [0033]. As mapped, the “computing environment” is the customer's cloud account, such as the customer's AWS or multi-cloud deployment. A POSITA would understand that a “machine” in the context of AWS or a “compute asset” in the context of cloud computing would refer to virtual machines. Ex. 1003, ¶¶ 87-89. Accordingly, Shua teaches *generating an inspectable disk based on an original disk of a resource deployed in a computing environment.*

3. Claim 1[b]: “inspecting the inspectable disk for an AI model”

Shua in view of Lang teaches *inspecting the inspectable disk for an AI model.* In particular, Shua teaches that, once the snapshot (“inspectable disk”) is created for

each of the customer's cloud assets, those snapshots undergo an analysis (inspecting) process to identify all of the customer's assets and build a "full-stack inventory":

Orca Security may involve a one-time, essentially instantaneous, impact-free integration into AWS, Azure, or GCP. Following its one-time integration, Orca scans the configuration, network layout, and security configuration. (This may be performed via a scanner account.) **It does so while also reading into virtual machines, disks, databases, and datastores**, as well as logs for cloud assets. **It then analyzes the data and builds a full-stack inventory.** Next it automatically assesses the security state of discovered asset throughout the technology stack, including all four cloud layers: I/S, OS, apps, and data.

Ex. 1005, *Shua Provisional*, [0030]. Shua emphasizes that "every asset in the account may be enumerated in all regions." *Id.*, [0035]; *see also* [0033] (Orca builds a map of the customer's "entire estate"); Ex. 1004, *Shua*, 9:53-94. Shua discusses analyzing the snapshot (i.e., inspecting the inspectable disk) for assets and services such as clusters, repositories, functions, and CloudFront services, and discusses "Vulnerability Scanning" (Ex. 1005, *Shua Provisional*, [0039]) but does not expressly disclose inspecting the inspectable disk for an AI model.

However, Lang does teach an analysis system for "extracting [] AI components" (*inspecting ... for an AI model*):

FIG. 32 depicts the **analysis system for extracting and reversing AI components from containers and/or virtual machines (VM)**. the (*sic*) container and/or **VM may be ingested by the analysis system**

and AI components are extracted, such as (but not limited to) analyzing files on the file system, reversing binaries and extracting AI components, detecting installed AI dependencies, analyzing AI dependency versions and metadata, etc.

Ex. 1006, *Lang*, [0331]; *see also* [0195] (“FIG. 12 illustrates how AI models may be detected, understood and/or reversed”); [0030]-[0032]. *Lang* expressly contemplates that this analysis system can operate on “VM files, VM hard disk files and/or any other data related to containers or VMs that may or may not contain AI information, and/or other information pertaining to containers or VMs.” *Id.* [0332]. A POSITA would understand that a snapshot of a virtual machine disk would be a “VM hard disk file[]” or at least “other data related to containers or VMs.” Ex. 1003, ¶ 95. Accordingly, *Lang* teaches a process of *inspecting the inspectable disk for an AI model*.

Furthermore, it would have been obvious to a POSITA at the time of the ’529 patent to modify the inspection of assets and services on an inspectable disk as in *Shua* to include the inspection specifically for an AI model as taught by *Lang*. Both *Shua* and *Lang* relate to security analyses of assets in a cloud environment, and a POSITA would be motivated to modify *Shua* to include AI systems in view of increased needs in the industry “to ensure AI systems are working as intended.” Ex. 1006, *Lang*, [0006]; *see also* Ex. 1003, ¶¶ 96-97. AI systems are “prone to adversarial attacks, wherein an attack manipulates the AI model or data (incl.

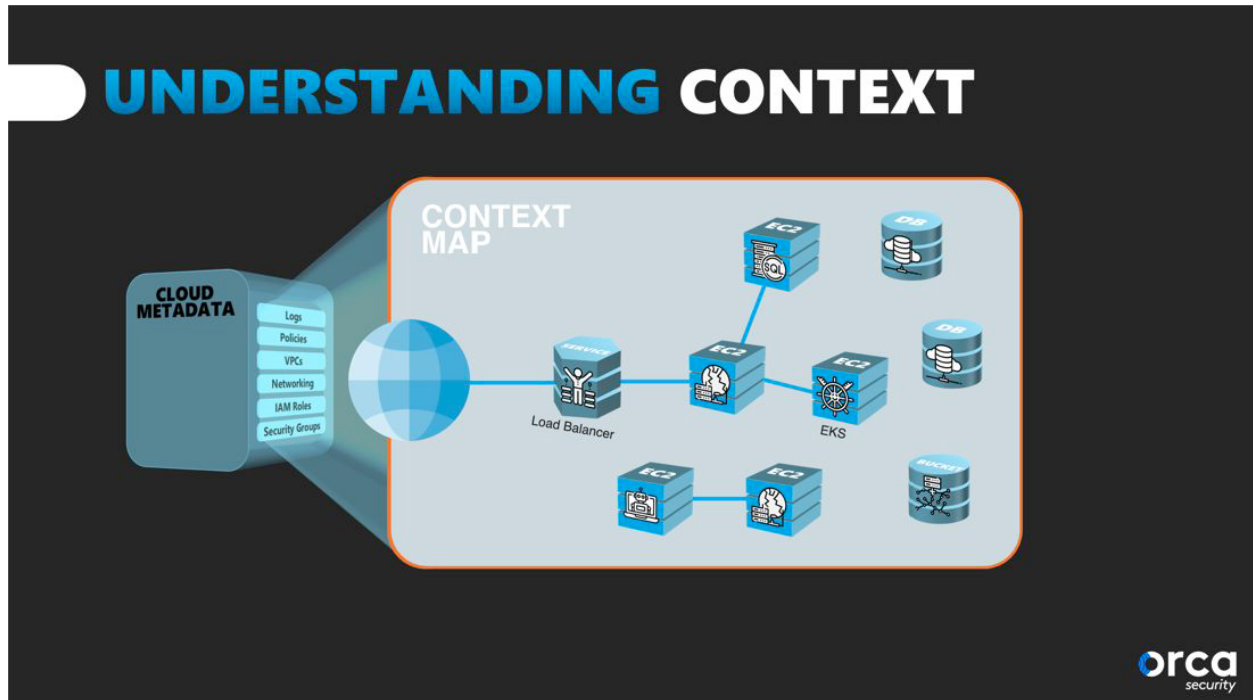
training or inference data) in a way that affects the model's behavior and decision-making.” *Id.* AI models are “highly proprietary” and “frequently provide information about crucial systems and their data and behavior,” and are thus “a target for theft.” *Id.* Accordingly, a POSITA would be motivated to assess cybersecurity risks for AI services (as taught by Lang) in Shua's system for assessing risks of assets and services in a cloud computing environment. Ex. 1003, ¶ 97. *See also* limitation 1[P], *supra*.

4. Claim 1[c]: “generating a representation of the AI model in a security database, the security database including a representation of the computing environment”

Shua in view of Lang teaches *generating a representation of the AI model in a security database, the security database including a representation of the computing environment*. The '529 Patent gives as an example of a security database, a “graph database, such as Neo4j®.” Ex. 1001, '529 Patent, 7:58-60. A POSITA would recognize that graph databases (including Neo4j) represent data as nodes; edges, which connect nodes to one another; and attributes of nodes and edges. Ex. 1003, ¶ 100 (citing Ex. 1018, *Graph Database Concepts*). Thus, in the terms of the '529 Patent, a stored representation of the computing environment as nodes, edges, and attributes of nodes and edges is a *security database including a representation of the computing environment*.

First, Shua teaches generating *a representation of the computing environment*.

As described above, Shua describes scanning the customer’s “entire environment” to “build a map” of it. Ex. 1005, *Shua Provisional*, [0033]. This map is represented (and subsequently visualized) as a graph:



Ex. 1007, *Orca Security SideScanning Technical Brief*⁴, p. 9; see also Ex. 1003, ¶¶ 67-68 (Dr. Zadok confirming that this figure corresponds to the reduced-quality version appearing at Ex. 1005, *Shua Provisional*, p. 15). This graph representation

⁴ For the sake of readability, the higher-quality versions of the images from Ex. 1007 are reproduced in this Petition. For the avoidance of doubt, Ex. 1007 does not form

includes nodes (representing systems and devices) and edges (“vectors”) representing connections between the nodes:

In step 227, scanning system 101 may generate a map of cloud infrastructure 106. In some embodiments, **this map may be in a form of graph: a plurality of interconnected vectors connecting the plurality of systems and devices**, based on the networking configuration.

Ex. 1004, *Shua*, 13:1-5. This graph is also annotated with attributes representing properties of the assets:

Generating the map in step 227 may include recording information such a region identifier, site identifier, datacenter identifier, physical address, network address, workload name, or any other identifier which may be acquired via an API provided through a cloud service provider's system.

Id., 13:36-40. Finally, a POSITA would understand that Shua’s graph representation is stored, at least because generating it includes “recording information” (*id.*) and because Shua contemplates comparing the original map to a “new map” generated subsequent to a change. *Id.* 11:3-20; *see also* Ex. 1003, ¶ 104. Accordingly, Shua teaches *generating a representation of* all of the entities in the customer’s computing

part of the Grounds for Unpatentability presented herein, and the figures relied upon as evidence of unpatentability are the lower-quality versions from Ex. 1005.

environment in a security database, the security database including a representation of the computing environment. Shua does not teach specifically generating a representation of the AI model as a part of this representation of the computing environment.

However, Lang teaches generating a representation of the identified model, which a POSITA would understand to have been stored with the other cloud asset representations, as set forth below. *See also* Ex. 1003, ¶ 105. First, Lang teaches generating different representations of the AI model located via inspection, including an “abstract AI model” suitable for import into another system (such as Shua’s system):

An abstract AI model 1260 module may process and/or abstract AI model functions into semantical representations for example to allow for easier import into a system... [¶] **Abstracting AI model functions into semantical representations may be implemented using approaches such as** (but not limited to): semantic reasoning about detected AI tasks, mathematical comparison between the run binaries and/or abstracted functions, comparison to known and/or common AI model designs and/or formats such as neural network architectures, and/or other **approaches to create an abstract representation of the AI model.** If abstraction is successful, **the output of an abstract AI model 1260 module may include an abstracted AI model, which may be written to a storage such as a storage device, and/or networking device, etc.**

Lang, [0210]-[0211]. Lang teaches the “abstract AI model 1260” step is performed as a part of FIG. 12, which “illustrates **how AI models may be detected**, understood and/or reversed ... **from** JIT, bytecode, assembly code, **a computer binary**, source code, full syntax trees, and/or abstract syntax trees, etc.” *Id.*, [0195]. A POSITA would understand that a computer binary containing AI code would be present in the virtual machine snapshots captured by Shua’s system. Ex. 1003, ¶¶ 105-106.

In the Shua-Lang system, Shua’s would capture a snapshot of a disk from a virtual machine in the customer’s cloud computing account, and Lang’s system would inspect files on the snapshot to identify the AI model (*see supra* limitation 1[b]) and generate an abstract representation of the AI model that can be stored in Shua’s map representing the customer’s computing environment.

In addition to the motivations to modify Shua’s system to incorporate Lang’s teaching discussed above (*see* 1[P] and [1b]), a POSITA would further be motivated to add Lang’s representations of the identified AI models into Shua’s map based on Shua’s teaching that the map should include the customer’s “entire environment,” at least because the AI models would represent cloud assets that are part of the customer’s “entire environment.” Ex. 1005, *Shua Provisional*, [0033]; Ex. 1003, ¶ 108.

5. Claim 1[d]: “inspecting the AI model for a cybersecurity risk”

Shua in view of Lang teaches *inspecting the AI model for a cybersecurity risk*.

In particular, Shua teaches inspecting assets, services, and programs for cybersecurity risks:

In **performing vulnerability assessment**, Orca may extract the OS packages, installed software applications, libraries, and program language libraries such as Java archives, Python packages, Go modules, Ruby gems, PHP packages, and Node.js modules, or any other software applications. It gathers library versions, software versions and other identifying characteristics, and **in a later phase tries matching them to known vulnerabilities in its vulnerability database** (e.g. data structure of known vulnerabilities).

Ex. 1005, *Shua Provisional*, [0042]; *see also id.* [0030]. Shua does not expressly disclose *inspecting the AI model*. However, Lang discloses analyzing AI components (i.e., inspecting the AI model) for a cybersecurity risk by, for example detecting poisoning in the data:

For example, **the analysis action may provide a variety of analyses and/or provide a variety of results. This may include**, but is not limited to, analyzing the accuracy of models, **detecting poisoning in data**, binary analysis, source code analysis, detecting adversarial attacks, assessing adversarial robustness, analyzing reliability, finding privacy violations, analyzing safety and/or security of the system, analyzing the system against compliance controls, detecting vulnerabilities and/or risks, evaluating the models against OOD data,

analyzing explainability (and transparency) aspects of the model, analyzing bias, analyzing fairness, analyzing flexibility, and/or testing the security of endpoints, etc.

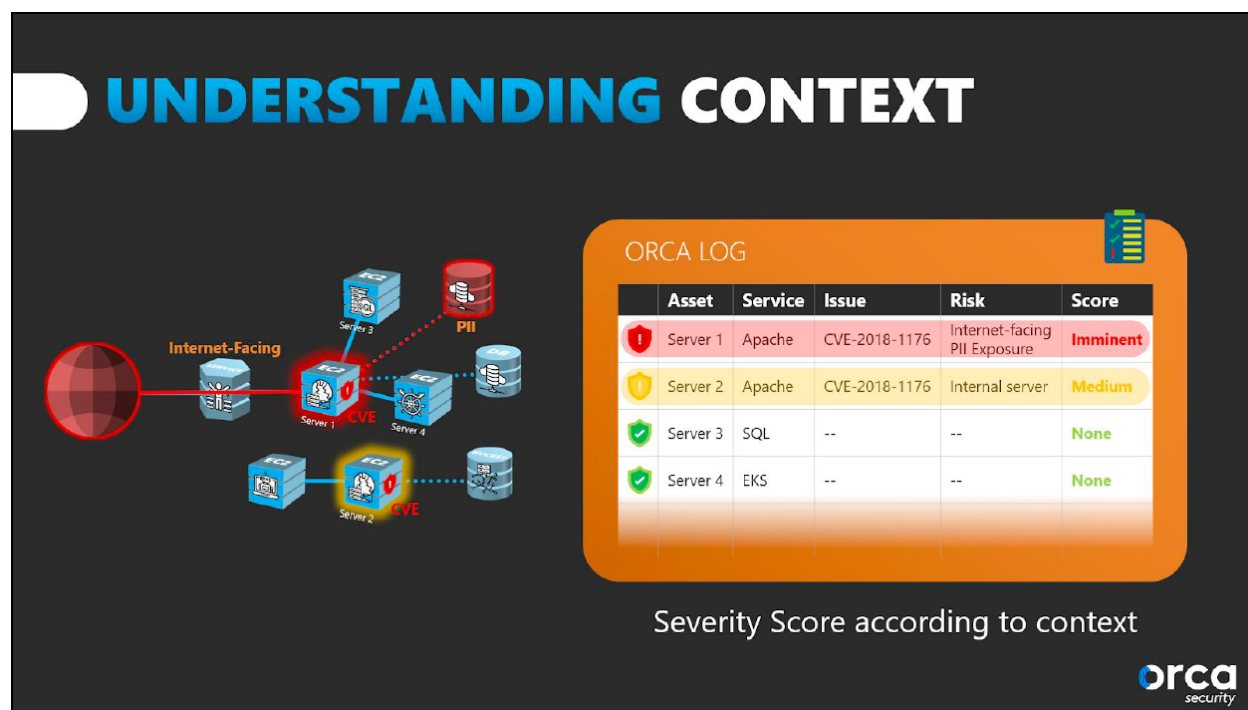
Lang, [0120]; *see also* [0270] (giving a “poisoning attack” as one example of “vulnerabilities and/or attacks” on AI models). Accordingly, Shua in view of Lang teaches *inspecting the AI model for a cybersecurity risk*.

In addition to the motivations to modify Shua’s system to incorporate Lang’s teaching discussed above (*see* 1[P] and [1b]), a POSITA would further be motivated to add Lang’s inspection of the identified AI models for cybersecurity risks into Shua’s map based on Shua’s teaching that the system should “identify risks 803 associated with each of the identified assets,” and would recognize that the AI models would be identified assets in the Shua-Lang system, and that Lang’s vulnerabilities and/or attacks would represent cybersecurity risks in those assets. Ex. 1004, *Shua*, 49:24-26; Ex. 1003, ¶¶ 114-115.

6. ***Claim 1[e]: “generating a representation of the cybersecurity risk in the security database, the representation of the cybersecurity risk connected to the representation of the AI model in response to detecting the cybersecurity risk”***

Shua in view of Lang teaches *generating a representation of the cybersecurity risk in the security database, the representation of the cybersecurity risk connected to the representation of the AI model in response to detecting the cybersecurity risk*.

In particular, Shua discloses *generating a representation of the cybersecurity risk in the cybersecurity database* at least because Shua teaches that representations of cybersecurity risks are added to its map (*security database*) to “contextualize assets and their associated risks.” Ex. 1005, *Shua Provisional*, [0036]. For example, Shua states that its system “**combines conclusions** from different environmental perspectives (e.g., metadata) **into a single model**. It takes the asset map from the control plane—where the service meshes and containers talk to each other—and **enriches this information with the risk data** gathered from the workload deep scanning.” *Id.* [0061]. This process is illustrated below:



Ex. 1007, *Orca Security SideScanning Technical Brief*, p. 17; *see also* Ex. 1003, ¶¶ 67-68 (Dr. Zadok confirming that this figure corresponds to the reduced-quality version appearing at Ex. 1005, *Shua Provisional*, p. 25).

Comparing this visualization of the map to the version depicted above with respect to limitation 1[c], the original map has been augmented with vulnerability data. Ex. 1003, ¶ 119 (Dr. Zadok noting that “Server 1” and “Server 2” have been annotated with a “CVE” attribute). A POSITA would understand that these attributes are each a “representation ... in the security database” because they are each an attribute of the corresponding node and are therefore stored as an attribute (*representation*) of the node (affected asset such as Server 1) in the graph database representing Shua’s map (*security database*). *Id.*, ¶ 120. Finally, a POSITA would understand that “CVE” stands for “Common Vulnerabilities and Exposures,” a well-known program to catalog publicly disclosed cybersecurity vulnerabilities, and that the “Issue” annotations in the “Orca Log” are consistent with CVE identifiers. *Id.*, ¶¶ 119-120.

Next, these representations are added to Shua’s map *in response to detecting the cybersecurity risk*. As described above, Shua’s system “takes the asset map ... and enriches this information with **the risk data gathered from the workload deep scanning.**” Ex. 1005, *Shua Provisional*, [0061]. This data is stored in the map after (in response to) detecting the risks in order to present them (as depicted above):

Orca detects risks such as vulnerabilities, malware, misconfiguration, and lateral movement risk in the cloud. It sees what other security tools miss. **Following data gathering and analysis, Orca reports all issues in full context**, including where each risk resides and its true level of risk. This lets its customers prioritize their mitigation actions while mitigating cloud risk.

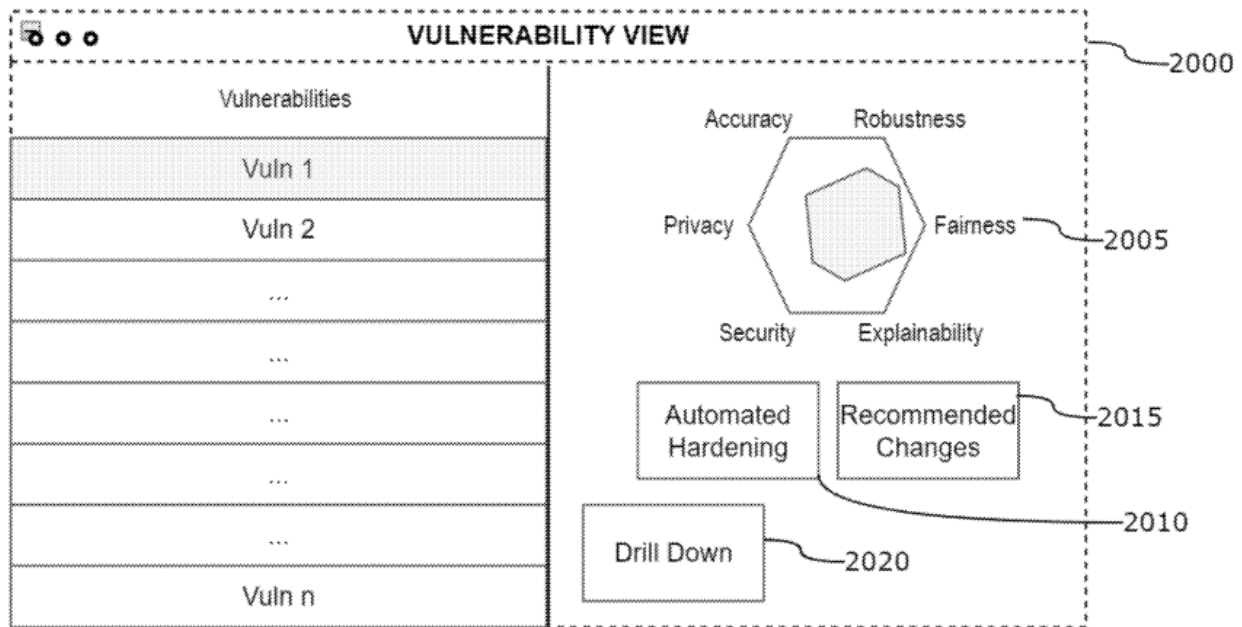
Id., [0070]. Furthermore, Shua discloses that a new map is generated “based on” detected changes in security-related state (e.g., trust-related changes) and compared to the previous map. Ex. 1004, *Shua*, 11:3-20. A POSITA would understand that the new map would be created (and the vulnerability attribute representations added) “responsive to” detecting a new vulnerability. Ex. 1003, ¶¶ 121-122.

Accordingly, A POSITA would understand that Shua teaches *generating a representation (the annotation) of the cybersecurity risk (the vulnerability identified by the particular CVE identifier) in the security database (the graph database representing Shua’s map) ... connected to the representation of the cloud computing asset (stored as an attribute of the affected node) in response to detecting the cybersecurity risk.*

Shua discloses connecting assets, services, and programs (e.g., models) with cybersecurity risks as described above, but does not expressly describe connecting an AI model with a cybersecurity risk. However, Shua broadly discloses storing cybersecurity risk attributes for an asset in association with the node representing

that asset. Similarly, Lang teaches generating a representation of an identified vulnerability and associating identified vulnerabilities for a model with that model.

For example, Lang discloses the user interface of FIG. 20:



Ex. 1006, *Lang*, FIG. 20. A POSITA would understand that the representations of vulnerabilities depicted above (Vuln 1, Vuln 2, etc.) would be stored “connected to” (e.g. linked to) the representation of the model, at least to enable the depicted “automated hardening,” “recommend changes,” and “drill down” functions. Ex. 1003, ¶¶ 124-125.

In addition to the motivations to modify Shua’s system to incorporate Lang’s teachings discussed above (*see* 1[P] and [1b]), a POSITA would further be motivated to add Lang’s associations of AI models with associated cybersecurity risks in Shua’s map based on Shua’s teaching that the system should “contextualize assets

and their associated risks discovered within the customer cloud account,” including AI assets. Ex. 1005, *Shua Provisional*, [0036]; Ex. 1003, ¶ 126.

7. ***Claim 1[f]: “initiating a mitigation action based on the cybersecurity risk”***

Shua teaches *initiating a mitigation action based on the cybersecurity risk*. In particular, Shua teaches mitigating the discovered vulnerabilities (cybersecurity risks):

Upon gathering network accessibility information and the identified port to identify one or more vulnerabilities susceptible to attack from outside the workload, **a disclosed embodiment would have the functionality** to perform processes to gather, display, and **mitigate a discovered vulnerability in order to minimize the likelihood and effectiveness of a cyber threat** outside of and attempting to access a workload through a known or previously encountered type of cyber threat. **This functionality may include** collecting and organizing the vulnerabilities according to type or category of vulnerability, displaying the gathered data for an end user or maintainer, and **implementing security features automatically** or manually by a user or maintainer such as security patches, password or passcode changes and suggestions for users to do the same, and malicious code eradication.

Ex. 1004, *Shua*, 24:61-25:9; *see also id.* 30:31-33. To the extent Patent Owner contends that this limitation requires the mitigation to be a mitigation of a

cybersecurity risk in an AI model, Lang teaches mitigating security vulnerabilities in AI models:

Analysis and/or hardening may work closely with one or more monitoring actions. **For example, monitoring may include continuously analyzing the AI systems 105 for vulnerabilities. If vulnerabilities are discovered, then a hardening action may be used to harden the system against those vulnerabilities.** Monitoring may include continuously assessing logs, inputs, and/or outputs, etc., of the model. **It may include reporting on any issues, and/or it may include making mitigations.** Mitigations may include, but are not limited to, blocking access to specific components of the systems, shutting down entire systems, blocking access by a specific user, blocking specific types of inputs, emailing users, texting users, alert users, and/or trigger a siren, etc.

Ex. 1006, Lang, [0122]. Accordingly, a POSITA would understand that Lang teaches AI model-specific mitigations. Ex. 1003, ¶¶ 130-131.

In addition to the motivations to modify Shua's system to incorporate Lang's teachings discussed above (*see* 1[P] and [1b]), a POSITA would further be motivated to add Lang's teaching of mitigating vulnerabilities in AI models into Shua's more general teachings of mitigating vulnerabilities based on Shua's teaching that mitigating vulnerabilities is desirable. Ex. 1004, *Shua*, 64:36-43; Ex. 1003, ¶ 132. A POSITA would additionally understand that mitigating identified vulnerabilities is desirable to improve security in cloud-based systems and would be motivated to

incorporate Lang's mitigation teaching into Shua's system based on that understanding as well. *Id.* ¶ 132.

B. Claim 2

1. *Claim 2: "detecting an artifact of the AI model on the inspectable disk"*

Shua in view of Lang teaches *detecting an artifact of the AI model on the inspectable disk*. The '529 patent explains that an artifact of an AI model is "a file, a software library, a software binary, a folder name, various combinations thereof, and the like, which indicate that a known (e.g., previously determined) AI model is installed on a disk, resource, and the like, which is being inspected." Ex. 1001, '529 patent at 13:24-32. For example, the inspector of the '529 patent is "configured to detect a PyTorch framework." *Id.* at 13:33-34.

In particular, Shua teaches detecting artifacts of cloud assets as a part of the process of inspecting the inspectable disk:

In performing vulnerability assessment, Orca may extract the OS packages, installed software applications, **libraries**, and program language libraries such as Java archives, **Python packages**, Go modules, Ruby gems, PHP packages, and Node.js modules, or any other software applications. It gathers library versions, software versions and other identifying characteristics, and in a later phase tries matching them to known vulnerabilities in its vulnerability database (e.g. data structure of known vulnerabilities).

Ex. 1005, *Shua Provisional*, [0042]. *See also* limitation 1[b], *supra*. A POSITA would understand that PyTorch (given by the '529 Patent as an example of an artifact of an AI model) is a Python package that provides software libraries suitable for AI models. Ex. 1003, ¶ 136 (citing the PyTorch github repository). Accordingly, Shua would *detect an artifact ... on the inspectable disk*.

Shua does not expressly teach that the artifact detected is *an artifact of the AI model*. However, as a part of the disclosed model detection process, Lang expressly teaches detecting artifacts of AI models, including the same PyTorch framework given as an example by the '529 Patent:

A detect always present model functions 1315 module may scan an input computer binary 1305 (potentially received from a detect model headers and or file extensions 1310 module) for other data and/or functions that are always present in known AI formats. **Such scanning may for example (but not limited to) include scanning for known preprocessing functions that allow the binary to set up, known importing functions such as library imports that describe known AI libraries such as SciKit Learn, TensorFlow, PyTorch, Caffe, CNTK, MXNet, H2O, Core ML, Shogun, etc., and/or other known AI frameworks, and/or otherwise known functions that are always present for known AI frameworks, etc.**

Ex. 1006, Lang, [0226]. Lang further discloses that such artifacts are detected on an *inspectable disk* (i.e., the disk captured from a virtual machine). *Id.* [0029]-[0030]

(“Models may be analyzed in a consistent manner; detecting and/or analyzing AI components in virtual machines and containerized and virtual environments;”).

In addition to the motivations to modify Shua’s system to incorporate Lang’s teachings discussed above (*see* 1[P] and [1b]), a POSITA would further be motivated to add Lang’s teaching of collecting artifacts of AI models specifically into Shua’s process of collecting software libraries more generally based on Shua’s teaching that all software libraries should be collected to perform vulnerability analysis for those packages. *See* Ex. 1005, *Shua Provisional*, [0042]; Ex. 1003, ¶ 139.

C. Claim 3

1. *Claim 3: “inspecting the AI model to detect an AI model configured to execute a code object”*

Shua in view of Lang teaches *inspecting the AI model to detect an AI model configured to execute a code object*. First, Lang teaches *inspecting the AI model*. *See* limitation 1[d], *supra*. Lang further teaches that the AI models are inspected for preprocessing or importing functions that allow the AI model to import library code objects for execution:

A detect always present model functions 1315 module may scan an input computer binary 1305 (potentially received from a detect model headers and or file extensions 1310 module) for other data and/or functions that are always present in known AI formats. **Such scanning may for example (but not limited to) include scanning for known preprocessing functions that allow the binary to set up, known**

importing functions such as library imports that describe known AI libraries such as SciKit Learn, TensorFlow, PyTorch, Caffe, CNTK, MXNet, H2O, Core ML, Shogun, etc., and/or other known AI frameworks, and/or **otherwise known functions** that are always present for known AI frameworks, etc.

Ex. 1006, Lang, [0226]. A POSITA would appreciate that software libraries include executable code objects, and that software that imports such software libraries is configured to execute a code object embodied in the software library. Ex. 1003, ¶ 143.

In addition to the motivations to modify Shua's system to incorporate Lang's teachings discussed above (*see* 1[P] and [1b]), a POSITA would further be motivated to add Lang's teaching of inspecting AI models configured to execute code objects specifically into Shua's process of inspecting cloud assets more generally based on Shua's teaching that cloud assets should be inspected to perform vulnerability analysis for those packages and Lang's teaching that "source code analysis" in a part vulnerability analysis for AI models. *See Ex. 1005, Shua Provisional, [0042]; Ex. 1006, Lang, [0120]; Ex. 1003, ¶ 144.*

D. Claim 4

1. Claim 4: "detecting a metadata of the AI model, wherein the metadata indicates that the AI model is a cybersecurity risk"

Shua in view of Lang teaches *detecting a metadata of the AI model, wherein the metadata indicates that the AI model is a cybersecurity risk*. In particular, when

Lang discusses analyzing AI components from virtual machines, Lang expressly discloses that metadata of the AI model is extracted:

FIG. 32 depicts the **analysis system for extracting and reversing AI components from** containers and/or **virtual machines (VM)**. the (*sic*) container and/or VM may be ingested by the analysis system and **AI components are extracted, such as (but not limited to)** analyzing files on the file system, reversing binaries and extracting AI components, detecting installed AI dependencies, **analyzing AI dependency versions and metadata, etc.**

Ex. 1006, *Lang*, [0331]. Lang further describes how metadata is used in the analysis of AI models, including functional analysis:

A **detected AI model functions 1445 module may include functions related to AI**, including for example aggregated functions of all functions from a computer binary related to AI (incl. e.g., AI processes, and/or tasks, etc. etc. **This may include** one or more of binary, disassembled, semantic representations, **metadata**, and/or reports etc., and/ or may be a file such as a file (e.g., JSON, CSV, XML file, binary file, and/or code file etc.), which may be stored on a storage (e.g., physical media) and/or communicated over a network (e.g., via API).

Id. [0241]. Finally, Lang describes how metadata (such as software library version) may indicate that a particular AI model is a cybersecurity risk:

The analysis system may include original and permuted code segments that are stored in a source code segments 2820 database. **This database may include metadata**, such as (but not limited to) the

language, **version numbers of libraries used**, and/or dates, etc.

Source code segments may be used in a variety of analyses, for example (but not limited to), they may be used for: **vulnerability analysis where each segment is assessed for vulnerabilities**; similarity analysis to determine similar code segments that may be vulnerable; and/or testing the overall analysis system etc.

Id. [0302]; *see also id.* [0308] (particular model architectures or AI libraries may indicate that the AI model has a vulnerability and is therefore a cybersecurity risk). Accordingly, Shua in view of Lang teaches *detecting a metadata of the AI model* and that *the metadata indicates that the AI model is a cybersecurity risk*.

In addition to the motivations to modify Shua's system to incorporate Lang's teachings discussed above (*see* 1[P] and [1b]), a POSITA would further be motivated to add Lang's teaching of inspecting metadata of AI models specifically into Shua's process of inspecting cloud assets more generally based on Shua's teaching that cloud assets should be inspected to perform vulnerability analysis for those packages and Lang's teaching that metadata analysis in a part vulnerability analysis for AI models. *See* Ex. 1005, *Shua Provisional*, [0042]; Ex. 1006, *Lang*, [0302], [0308]; Ex. 1003, ¶ 150.

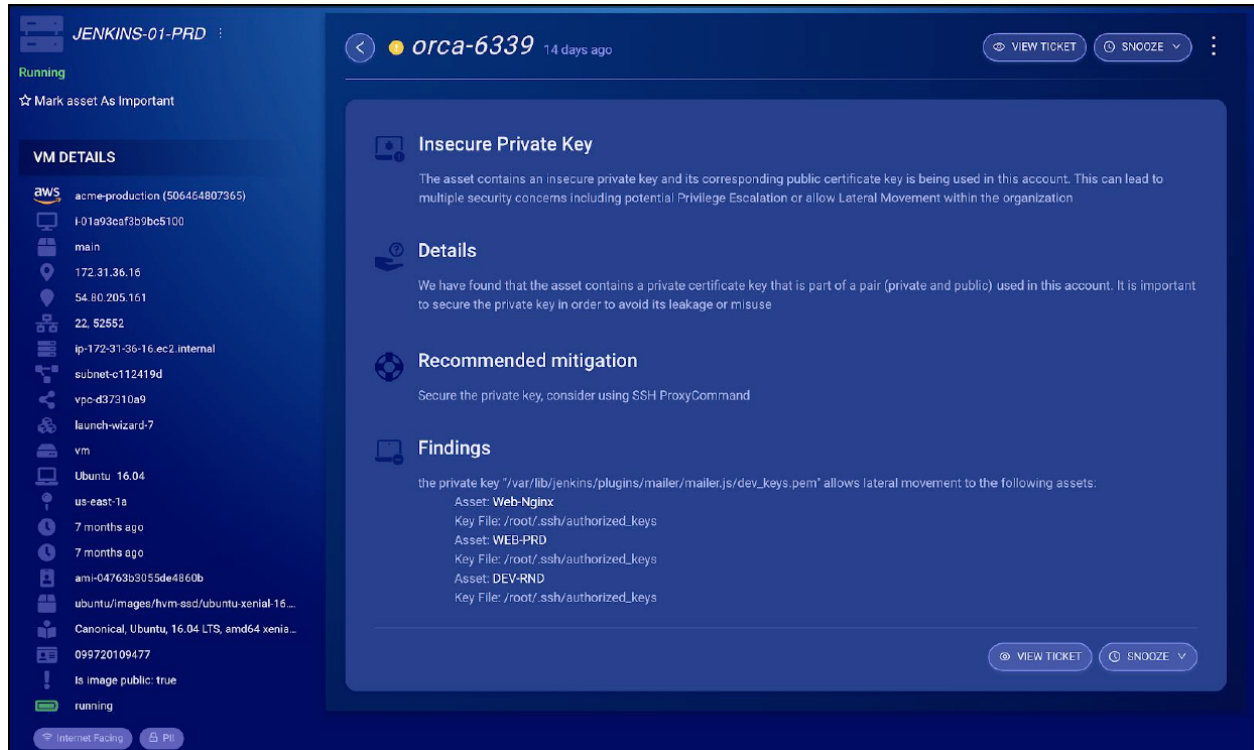
E. Claim 5

1. Claim 5: “detecting in the AI model any one of: a secret, a certificate, a code, and any combination thereof”

Shua in view of Lang teaches *detecting in the AI model ... a secret* and *detecting in the AI model ... a certificate*. The ‘529 Patent gives, as an example of a “secret,” a credential “utilized by an administrator account to access a storage in a cloud computing environment,” which may therefore enable a lateral movement path for an adversary. Ex. 1001, ‘529 Patent, 14:62-67. A POSITA would understand that such credentials can take the form of passwords or (cryptographic) private keys. Ex. 1003, ¶ 153.

First, Shua teaches detecting credentials enabling lateral movement (such as private keys, passwords) in the inspectable disk:

Orca Security scans your entire cloud account in a holistic manner. **It peers into each machine's filesystem for private keys**, such as on server A, taking a hash of each found. It then checks the authorized key configuration across all other assets for public keys. If any are found, such as on server B, Orca calculates hashes and compares the private key hash to find a match.



From the image above we see that Orca provides: The path to the insecurely stored key: /var/lib/jenkins/plugins/mailler/mailler.js/dev_keys.pem. A recommended solution. The identity of the machines the key allows access to e.g., (Web-Nginx, WEBPRD, DEV-RND). The user account on the machine in this case (root user for all machines in this example). **Other lateral movement risks Orca finds are: Passwords in Shell History. Sensitive Info in Git Repository. Sensitive AWS Keys on Systems.**

Ex. 1005, *Shua Provisional*, [0085]-[0086]; Ex. 1007, *Orca Security SideScanning Technical Brief*, p. 13; see also Ex. 1003, ¶¶ 67-68 (Dr. Zadok confirming that this figure corresponds to the reduced-quality version appearing at Ex. 1005, *Shua Provisional*, p. 30).

Next, Shua teaches detecting “electronic certificates” enabling trust between a customer’s cloud assets:

Another embodiment contemplated may comprise a method for identifying relationships 305 between at least some of the identified assets 801, the relationships including at least one of a trust 807A, a network connectivity 807B, or a mechanism of network proxying 807C. A trust 807A may establish a relationship and may involve cryptography, digital signatures, electronic certificates, or similar method of establishing a trusting relationship.

Ex. 1004, *Shua*, 49:36-43. Accordingly, Shua teaches *detecting ... a secret, detecting ... a certificate*. Shua does not, however, expressly teach detecting a secret or a certificate *in the AI model*.

However, Lang does teach analyzing components of AI (including files used by AI components) in the aid of vulnerability analysis:

The analysis system may include extracted AI information 3240 that represents output of an of the analysis system depicted in FIG. 32 for analyzing containers and VMs. **This may include, but is not limited to, AI components, a report of detected AI, a bill of materials of AI components, a list of directory and file paths to AI components, reversed AI components, and/or source code segments, etc. Such output may then be used by the analysis system (or elsewhere), for example for (but not limited to) source code analysis, adversarial attack analysis, and/or bias analysis, etc.**

Ex. 1006, *Lang*, [0339]. A POSITA would appreciate that the Shua-Lang system, when scanning each machine's filesystem (as taught by Shua) and examining the list of file paths used by AI model (as taught by Lang), would detect the secrets and certificates expressly scanned for by Shua's system when they are a component used by the AI model, at least because Lang teaches identifying all file paths "to AI components." *Id.* Ex. 1003, ¶¶ 156-160. As such, Shua in view of Lang teaches *detecting in the AI model ... a secret and detecting in the AI model ... a certificate.*

In addition to the motivations to modify Shua's system to incorporate Lang's teachings discussed above (*see* 1[P] and [1b]), a POSITA would further be motivated to add Lang's teaching of searching AI model file and directory paths into Shua's process of inspecting each machine's filesystem for sensitive information at least based on Shua's teaching that "it's essential to know where sensitive information, such as insecurely stored keys, is located across your entire cloud environment." *See* Ex. 1005, *Shua Provisional*, [0084]; Ex. 1003, ¶ 161.

F. Claim 6

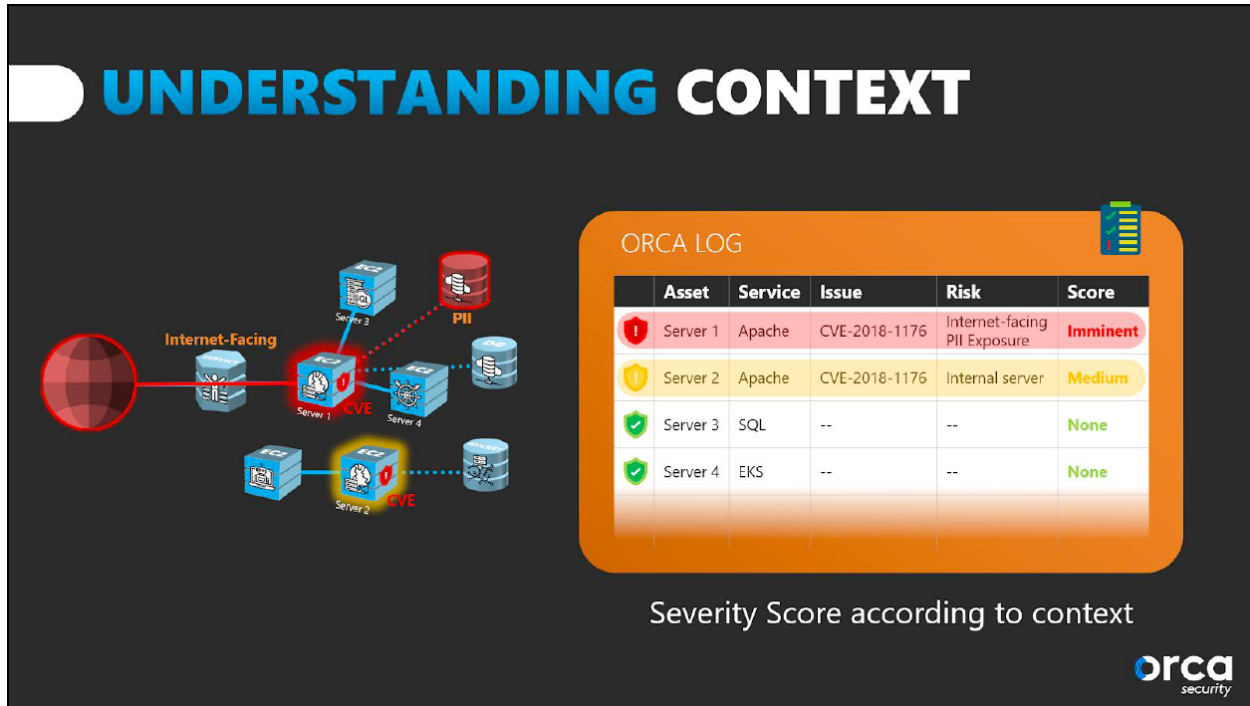
- 1. Claim 6: "generating a lateral movement path, wherein the lateral movement path includes the AI model, and a representation of a resource, wherein the resource is accessible utilizing the AI model"**

Shua in view of Lang teaches *generating a lateral movement path, wherein the lateral movement path includes the AI model, and a representation of a resource, wherein the resource is accessible utilizing the AI model.*

In particular, Shua teaches *generating a lateral movement path, wherein the lateral movement path includes the [asset or service], and a representation of a resource, wherein the resource is accessible utilizing the [asset or service]* via its detection of lateral movement risk and its map representing cloud resources (databases with exposed PII) accessible using other assets and services with connections to risks (e.g., CVEs).

For example, based on the contextual map, the system may perform a forward analysis of the specific asset under identification to identify at least one possible internet-originating attack vector to the asset. Alternatively or additional, **the system may perform a backward analysis of the specific asset to identify exposure risk to assets downstream of the specific asset, wherein the downstream exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and lateral movement risks associated with the exposed asset.** Such analysis may be presented to a user via graphical user interface.

Ex. 1005, Shua Provisional, [0061]. Shua expressly shows this “lateral movement path,” as presented to the user:



Ex. 1007, *Orca Security SideScanning Technical Brief*, p. 17; see also Ex. 1003, ¶¶ 67-68 (Dr. Zadok confirming that this figure corresponds to the reduced-quality version appearing at Ex. 1005, *Shua Provisional*, p. 25). Shua’s depiction shows the lateral movement path in red, from the public internet, via “Server 1” with a CVE, to the exposed database with PII. As discussed above, Server 1 and the exposed database each have a corresponding *representation of a resource* in Shua’s map. As such, Shua alone teaches *generating a lateral movement path* (shown in red above) *wherein the lateral movement path includes the [asset or service] (Server 1), and a representation of a resource* (the database with exposed PII), *wherein the resource is accessible utilizing the [asset or service]* (as shown in the lateral movement path).

Shua does not teach expressly that the lateral movement path includes *the AI model* or that the resource is accessible using *the AI model*.

However, Lang teaches that it is well-known that AI models include vulnerabilities, and that these vulnerabilities can allow unauthorized access to resources:

There has been a significant increase in research into the risks and vulnerabilities of AI systems, which include adversarial attacks that attempt to trick the system into revealing proprietary data or affecting its decision-making. These attacks often include small perturbations to input data that are difficult for a human to detect but drastically affect the AI system's decision-making process. An attack can take place before, during or after training, and before, during or after inference. For example, an attacker may introduce malicious data to the training data to introduce a backdoor to the system, reduce the model's accuracy, introduce biases, etc.

Ex. 1006, *Lang*, [0010].

As such, a POSITA would understand that, in the Shua-Lang system, the process of lateral path detection would include detecting lateral paths that include AI models. Ex. 1003, ¶¶ 166-168. For example, the Shua-Lang system can detect passwords or private keys in virtual machine AI models. *See* claim 5, *supra*. Shua expressly teaches that such passwords or private keys are lateral movement risks. Ex. 1005, *Shua Provisional*, [0086]. A POSITA would understand that in the Shua-

Lang system, once a lateral movement risk in an AI model is identified by scanning the AI model file paths per the teaching of Lang, the step of *generating a lateral movement path* would proceed per the teaching of Shua even when *the lateral movement path includes the AI model*. Ex. 1003, ¶ 169.

Furthermore, as discussed above, Lang teaches that AI model vulnerabilities grant unauthorized access to private resources. Ex. 1006, *Lang*, [0010]. Accordingly, a POSITA would understand that the Shua-Lang system would *generat[e] a lateral movement path, wherein the lateral movement path includes the AI model, and a representation of a resource, wherein the resource is accessible utilizing the AI model*.

In addition to the motivations to modify Shua's system to incorporate Lang's teachings discussed above (*see* 1[P] and [1b]), a POSITA would further be motivated to include lateral movement paths including AI model vulnerabilities (as identified per Lang's teachings) into Shua's process generating a master map putting all vulnerabilities in context (including the lateral movement risk associated with each vulnerability) at least based on Shua's teaching that "All such context [including vulnerabilities] is merged into the 'master map' discussed earlier, producing a map that includes the relationships between all of the workloads, discrete and containerized alike." *See* Ex. 1005, *Shua Provisional*, [0067]; Ex. 1003, ¶ 170.

G. Claim 7

1. Claim 7: “applying a policy on the representation of the computing environment”

Shua teaches *applying a policy on the representation of the computing environment*. The '529 patent explains that a “policy” might be “a rule, a conditional rule, a plurality of each, a combination thereof, and the like, which [is] applied to the representation of the computing environment, the representation of the AI model, a representation of a cybersecurity object, a combination thereof, and the like.” Ex. 1001, '529 Patent, 15:20-25. The '529 Patent goes on to give, as an example of a policy, “a rule to detect a toxic combination. For example, in an embodiment, a policy includes a rule applied to the representation to detect a finding [i.e., a detected cybersecurity risk] connected to a resource, and an AI model connected to the resource” *Id.*, 17:15-19. Thus, a “policy” in the language of the '529 Patent includes at least a rule to detect a configuration of resources that includes a vulnerability condition (“the toxic combination”).

Shua teaches such rules to detect a configuration of resources that includes a vulnerability condition:

Orca combines conclusions from different environmental perspectives (e.g., metadata) into a single model. It takes the asset map from the control plane-where the service meshes and containers talk to each other-and enriches this information with the risk data gathered from the workload deep scanning. **For example, Orca takes a customer's**

instance, maps the running services on it, then takes into consideration the vulnerability data collected in the data path. Orca contextualizes the information to establish whether an asset is internet-facing and easily accessible to attackers, or if it's private and hence a less important vulnerability. ... [¶] Consider a vulnerability in a web service. Orca scores it as: High risk if it's connected to the internet, either directly or indirectly via a load balancer or reverse proxy; Medium severity if it's only accessible internally; Low severity if it's blocked by a security group configuration of the cloud provider.

Ex. 1005, *Shua Provisional*, [0061]. Shua thus includes a *policy* that detects a configuration of resources (the web service and the load balancer/reverse proxy) that includes a vulnerability condition (the vulnerability in the web server). Shua teaches *applying* this policy (rule) on the map (*representation of the computing environment*) to identify and categorize the configuration as high, medium, or low severity. Accordingly, Shua teaches *applying a policy on the representation of the computing environment*.

H. Claim 8

1. Claim 8: “initiating the mitigation action based on the policy”

Shua teaches *initiating the mitigation action based on the policy*. As discussed above, the '529 Patent teaches that a “policy” includes at least a rule to detect a configuration of resources that includes a vulnerability condition (the “toxic combination”). See claim 7, *supra*. The '529 Patent further specifies that mitigation

actions may be triggered by applying a rule. *Id.* 17:20-23 (“In an embodiment, **a mitigation action is initiated based on a toxic combination**, the detected cybersecurity object, an AI model type, a combination thereof, and the like.”).

Like the ’529 Patent, Shua specifies that policies, when applied, can trigger mitigation actions:

Upon gathering network accessibility information and the identified port to identify one or more vulnerabilities susceptible to attack from outside the workload, a disclosed embodiment would have the functionality to perform processes to gather, display, and mitigate a discovered vulnerability in order to minimize the likelihood and effectiveness of a cyber threat outside of and attempting to access a workload through a known or previously encountered type of cyber threat.

Ex. 1004, *Shua*, 24:61-25:2; *see also id.* 31:12-27, 32:25-48. Accordingly, Shua teaches *initiating the mitigation action based on the policy*.

I. Claim 9

- 1. Claim 9[P]: “non-transitory computer-readable medium storing a set of instructions for detecting a cybersecurity risk of an artificial intelligence (AI), the set of instructions comprising: one or more instructions that, when executed by one or more processors of a device, cause the device to”**

To the extent the preamble is limiting, Shua teaches *non-transitory computer-readable medium storing a set of instructions for detecting a cybersecurity risk of*

an artificial intelligence ... that, when executed by one or more processors of a device, perform the limitations discussed below.

In particular, Shua teaches computer-readable media storing instructions to perform Shua's disclosed techniques:

Embodiments described herein may refer to **a non-transitory computer readable medium containing instructions that when executed by at least one processor, cause the at least one processor to perform a method.** Non-transitory computer readable medium may include any medium capable of storing data in any memory in a way that may be read by any computing device with a processor to carry out methods or any other instructions stored in the memory.

Ex. 1004, *Shua*, 4:48-55. Lang similarly teaches computer-readable media storing instructions to perform Lang's disclosed techniques:

Further, many examples are described in terms of sequences of actions to be performed by, for example, elements of a computing device. It will be recognized that various actions described herein can be performed by specific circuits (e.g., Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGA), Graphics Processing Units (GPU)), by program instructions being executed by one or more processors, or by a combination of both. Additionally, **these sequences of actions described herein can be considered to be embodied entirely within any form of computer readable storage medium having stored therein a corresponding set of computer**

instructions that upon execution would cause an associated processor to perform the functionality described herein.

Ex. 1006, *Lang*, [0076].

See also limitation 1[P], *supra*.

2. ***Claim 9[a]: “generate an inspectable disk based on an original disk of a resource deployed in a computing environment”***

See limitation 1[a], *supra*.

3. ***Claim 9[b]: “inspect the inspectable disk for an AI model”***

See limitation 1[b], *supra*.

4. ***Claim 9[c]: “generate a representation of the AI model in a security database, the security database including a representation of the computing environment”***

See limitation 1[c], *supra*.

5. ***Claim 9[d]: “inspect the AI model for a cybersecurity risk”***

See limitation 1[d], *supra*.

6. ***Claim 9[e]: “generate a representation of the cybersecurity risk in the security database, the representation of the cyber-security risk connected to the representation of the AI model in response to detecting the cybersecurity risk”***

See limitation 1[e], *supra*.

7. ***Claim 9[f]: “initiate a mitigation action based on the cybersecurity risk”***

See limitation 1[f], *supra*.

J. Claim 10

1. ***Claim 10[P]: “A system for detecting a cybersecurity risk of an artificial intelligence (AI) comprising: a processing circuitry; a memory, the memory containing instructions that, when executed by the processing circuitry, configure the system to”***

To the extent the preamble is limiting, Shua discloses a *system ... comprising a processing circuitry, a memory, the memory containing instructions that, when executed by the processing circuitry* perform the limitations discussed below.

In particular, Shua discloses a “system” comprising “a memory” and “instructions” executed by “processing circuitry”:

The application program may be uploaded to, and executed by, a machine having any suitable architecture. Preferably, the machine may be implemented on a **computer platform having hardware such as one or more central processing units ("CPUs"), a memory, and input/output interfaces. The computer platform may also include an operating system and microinstruction code. The various processes and functions described in this disclosure may be either part of the microinstruction code or part of the application program, or any combination thereof, which may be executed by a CPU**, whether or not such a computer or processor is explicitly shown.

Ex. 1004, *Shua*, 4:62:-5:6. Lang similarly teaches a system including “one or more processors” and “computer readable storage medium having stored therein a corresponding set of computer instructions”:

Further, many examples are described in terms of sequences of actions to be performed by, for example, elements of a computing device. It will be recognized that various actions described herein can be performed by specific circuits (e.g., Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGA), Graphics Processing Units (GPU)), by program instructions being executed by one or more processors, or by a combination of both. Additionally, **these sequences of actions described herein can be considered to be embodied entirely within any form of computer readable storage medium having stored therein a corresponding set of computer instructions that upon execution would cause an associated processor to perform the functionality described herein.**

Ex. 1006, *Lang*, [0076].

See also limitation 1[P], *supra*.

2. ***Claim 10[a]: “generate an inspectable disk based on an original disk of a resource deployed in a computing environment”***

See limitation 1[a], *supra*.

3. ***Claim 10[b]: “inspect the inspectable disk for an AI model”***

See limitation 1[b], *supra*.

4. ***Claim 10[c]: “generate a representation of the AI model in a security database, the security database including a representation of the computing environment”***

See limitation 1[c], *supra*.

5. Claim 10[d]: “inspect the AI model for a cybersecurity risk”

See limitation 1[d], *supra*.

6. Claim 10[e]: “generate a representation of the cybersecurity risk in the security database, the representation of the cyber-security risk connected to the representation of the AI model in response to detecting the cybersecurity risk”

See limitation 1[e], *supra*.

7. Claim 10[f]: “initiate a mitigation action based on the cybersecurity risk”

See limitation 1[f], *supra*.

K. Claim 11

1. Claim 11: “detect an artifact of the AI model on the inspectable disk”

See claims 10, 2 *supra*.

L. Claim 12

1. Claim 12: “inspect the AI model to detect an AI model configured to execute a code object”

See claims 10, 3 *supra*.

M. Claim 13

1. Claim 13: “detect a metadata of the AI model, wherein the metadata indicates that the AI model is a cybersecurity risk”

See claims 10, 4 *supra*.

N. Claim 14

1. Claim 14: “detect in the AI model any one of: a secret, a certificate, a code, and any combination thereof”

See claims 10, 5 *supra*.

O. Claim 15

- 1. Claim 15: “generate a lateral movement path, wherein the lateral movement path includes the AI model, and a representation of a resource, wherein the resource is accessible utilizing the AI model”**

See claims 10, 6 supra.

P. Claim 16

- 1. Claim16: “apply a policy on the representation of the computing environment”**

See claims 10, 7 supra.

Q. Claim 17

- 1. Claim17: “initiate the mitigation action based on the policy”**

See claims 16, 8 supra.

VI. CONCLUSION

For the foregoing reasons, Petitioner respectfully requests *inter partes* review of the Challenged Claims.

Respectfully submitted,

ERISE IP, P.A.

BY: /s/ Robin A. Snader

Robin A. Snader, Reg. No. 66,085
robin.snader@eriseip.com
7015 College Boulevard, Suite 700
Overland Park, Kansas 66211
(913) 777-5600 Telephone
(913) 777-5601 Facsimile

COUNSEL FOR PETITIONER

VII. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(A)(1)

A. Real Party-In-Interest

Petitioner is the only real party-in-interest. 37 C.F.R. § 42.8(b)(1).

B. Related Matters

The '693 Patent is presently asserted in the stayed district court litigation: *Orca Security Ltd. v. Wiz, Inc.*, No. 1:23-CV-00758 (D. Del.). The '529 Patent is not currently and has not been subject to any other district court litigation, *inter partes* review, reexamination, or other post-grant proceeding.

C. Lead and Back-Up Counsel

Petitioners provide the following designation and service information for lead and back-up counsel. 37 C.F.R. § 42.8(b)(3) and (b)(4).

Lead Counsel	Back-Up Counsel
Robin A. Snader (Reg. No. 66,085) robin.snader@eriseip.com PTAB@eriseip.com	Eric A. Buresh (Reg. No. 50,394) eric.buresh@eriseip.com
<u>Postal and Hand-Delivery Address:</u> ERISE IP, P.A. 7015 College Blvd., Suite 700 Overland Park, Kansas 66211 Telephone: (913) 777-5600 Fax: (913) 777-5601	Kathleen Fitterling (Reg. No 62,950) kathleen.fitterling@eriseip.com
	Christopher Schmidt (Reg. No. 63,982) chris.schmidt@eriseip.com
	<u>Postal and Hand-Delivery Address:</u> ERISE IP, P.A. 7015 College Blvd., Suite 700 Overland Park, Kansas 66211 Telephone: (913) 777-5600 Fax: (913) 777-5601

D. 37 C.F.R. § 42.8(b)(4) – Service Information

Please address all correspondence to the lead and back-up counsel as shown above. Petitioner consents to electronic service by e-mail at the e-mail addresses provided above.

CLAIM LISTING APPENDIX
U.S. Patent No. 12,003,529 for Claims 1-17

Claim Designation	Claim Language
Claim 1[P]	A method for detecting a cybersecurity risk of an artificial intelligence (AI), comprising:
Claim 1[a]	generating an inspectable disk based on an original disk of a resource deployed in a computing environment;
Claim 1[b]	inspecting the inspectable disk for an AI model;
Claim 1[c]	generating a representation of the AI model in a security database, the security database including a representation of the computing environment;
Claim 1[d]	inspecting the AI model for a cybersecurity risk;
	generating a representation of the cybersecurity risk in the security database, the representation of the cybersecurity risk connected to the representation of the AI model in response to detecting the cybersecurity risk; and
Claim 1[e]	initiating a mitigation action based on the cybersecurity risk.
Claim 2[P]	The method of claim 1, further comprising:
Claim 2[a]	detecting an artifact of the AI model on the inspectable disk.
Claim 3[P]	The method of claim 1, further comprising:
Claim 3[a]	inspecting the AI model to detect an AI model configured to execute a code object.
Claim 4[P]	The method of claim 1, further comprising:
Claim 4[a]	detecting a metadata of the AI model, wherein the metadata indicates that the AI model is a cybersecurity risk.
Claim 5[P]	The method of claim 1, further comprising:
Claim 5[a]	detecting in the AI model any one of: a secret, a certificate, a code, and any combination thereof.
Claim 6[P]	The method of claim 1, further comprising:
Claim 6[a]	generating a lateral movement path, wherein the lateral movement path includes the AI model, and a representation of a resource, wherein the resource is accessible utilizing the AI model.
Claim 7[P]	The method of claim 1, further comprising:
Claim 7[a]	applying a policy on the representation of the computing environment.
Claim 8[P]	The method of claim 7, further comprising:

Claim Designation	Claim Language
Claim 8[a]	initiating the mitigation action based on the policy.
Claim 9[P]	A non-transitory computer-readable medium storing a set of instructions for detecting a cybersecurity risk of an artificial intelligence (AI), the set of instructions comprising:
Claim 9[a]	one or more instructions that, when executed by one or more processors of a device, cause the device to:
Claim 9[b]	generate an inspectable disk based on an original disk of a resource deployed in a computing environment;
Claim 9[c]	inspect the inspectable disk for an AI model;
Claim 9[d]	generate a representation of the AI model in a security database, the security database including a representation of the computing environment;
Claim 9[e]	inspect the AI model for a cybersecurity risk;
Claim 9[f]	generate a representation of the cybersecurity risk in the security database, the representation of the cybersecurity risk connected to the representation of the AI model in response to detecting the cybersecurity risk; and
Claim 9[g]	initiate a mitigation action based on the cybersecurity risk.
Claim 10[P]	A system for detecting a cybersecurity risk of an artificial intelligence (AI) comprising:
Claim 10[a]	a processing circuitry;
Claim 10[b]	a memory, the memory containing instructions that, when executed by the processing circuitry, configure the system to:
Claim 10[c]	generate an inspectable disk based on an original disk of a resource deployed in a computing environment;
Claim 10[d]	inspect the inspectable disk for an AI model;
Claim 10[e]	generate a representation of the AI model in a security database, the security database including a representation of the computing environment;
Claim 10[f]	inspect the AI model for a cybersecurity risk;
Claim 10[g]	generate a representation of the cybersecurity risk in the security database, the representation of the cybersecurity risk connected to the representation of the AI model in response to detecting the cybersecurity risk; and
Claim 10[h]	initiate a mitigation action based on the cybersecurity risk.

Claim Designation	Claim Language
Claim 11[P]	The system of claim 10, wherein the memory contains further instructions which when executed by the processing circuitry further configure the system to:
Claim 11[a]	detect an artifact of the AI model on the inspectable disk.
Claim 12[P]	The system of claim 10, wherein the memory contains further instructions which when executed by the processing circuitry further configure the system to:
Claim 12[a]	inspect the AI model to detect an AI model configured to execute a code object.
Claim 13[P]	The system of claim 10, wherein the memory contains further instructions which when executed by the processing circuitry further configure the system to:
Claim 13[a]	detect a metadata of the AI model, wherein the metadata indicates that the AI model is a cybersecurity risk.
Claim 14[P]	The system of claim 10, wherein the memory contains further instructions which when executed by the processing circuitry further configure the system to:
Claim 14[a]	detect in the AI model any one of:
Claim 14[b]	a secret, a certificate, a code, and any combination thereof.
Claim 15[P]	The system of claim 10, wherein the memory contains further instructions which when executed by the processing circuitry further configure the system to:
Claim 15[a]	generate a lateral movement path, wherein the lateral movement path includes the AI model, and a representation of a resource, wherein the resource is accessible utilizing the AI model.
Claim 16[P]	The system of claim 10, wherein the memory contains further instructions which when executed by the processing circuitry further configure the system to:
Claim 16[a]	apply a policy on the representation of the computing environment.
Claim 17[P]	The system of claim 16, wherein the memory contains further instructions which when executed by the processing circuitry further configure the system to:
Claim 17[a]	initiate the mitigation action based on the policy.

APPENDIX OF EXHIBITS

Exhibit 1001	U.S. Patent No. 12,003,529 to Cohen et al.
Exhibit 1002	File History of the '529 Patent
Exhibit 1003	Declaration of Dr. Erez Zadok
Exhibit 1004	U.S. Patent No. 11,489,863 to Shua
Exhibit 1005	U.S. Provisional Patent App. No. 63/180,048 to Shua
Exhibit 1006	U.S. Patent Application Publication No. 2024/02022405A1 to Lang et al.
Exhibit 1007	Orca Security SideScanning Technical Brief, Orca Security, 2020
Exhibit 1008	Understanding Cloud Computing Basics (https://online.eou.edu/resources/article/understanding-cloud-computing-basics/), retrieved May 29, 2025
Exhibit 1009	The NIST Definition of Cloud Computing, NIST SP 800-145, September 2011
Exhibit 1010	NIST Cloud Computing Standards Roadmap, NIST SP 500-291, V2, July 2013
Exhibit 1011	What is Artificial Intelligence? (https://www.ibm.com/think/topics/artificial-intelligence), retrieved June 2, 2025
Exhibit 1012	What is Machine Learning? (https://www.ibm.com/think/topics/machine-learning), retrieved June 2, 2025
Exhibit 1013	NIST AI Risk Management Framework, NIST AI 100-1, January 2023
Exhibit 1014	What is Cybersecurity? (https://www.cisco.com/site/us/en/learn/topics/security/what-is-cybersecurity.html), retrieved May 29, 2025
Exhibit 1015	What is Cloud Security? (https://www.ibm.com/think/topics/cloud-security), retrieved June 2, 2025
Exhibit 1016	NIST Guide to Using Vulnerability Naming Schemes, NIST SP 800-51, R1, February 2011
Exhibit 1017	Graph Database for CyberSecurity (https://www.nebula-graph.io/posts/how-to-use-graphs-for-cybersecurity), retrieved June 2, 2025
Exhibit 1018	Neo4j Graph Database Concepts (https://neo4j.com/docs/getting-started/appendix/graphdb-concepts/), retrieved June 2, 2025

Exhibit 1019	About Archive and Standard Disk Snapshots (https://cloud.google.com/compute/docs/disks/snapshots), retrieved June 2, 2025
Exhibit 1020	PyTorch GitHub Page (https://github.com/pytorch/pytorch), retrieved June 2, 2025
Exhibit 1021	Joint Claim Construction Statement. <i>Orca Security Ltd. v. Wiz, Inc.</i> , No. 23-758, Dkt. 222 (D. Del.)

CERTIFICATION OF WORD COUNT

The undersigned certifies pursuant to 37 C.F.R. §42.24 that the foregoing Petition for *Inter Partes* Review, excluding any table of contents, mandatory notices under 37 C.F.R. §42.8, certificates of service or word count, or appendix of exhibits, contains 10,784 words according to the word-processing program used to prepare this document (Microsoft Word).

Dated: June 4, 2025

ERISE IP, P.A.

BY: /s/ Robin A. Snader
Robin A. Snader, Reg. No. 66,085
robin.snader@eriseip.com
7015 College Blvd., Suite 700
Overland Park, KS 66211
P: (913) 777-5600
F: (913) 777-5601

COUNSEL FOR PETITIONER

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. §§ 42.6(e) and 42.105(b), the undersigned certifies that on June 4, 2024, a complete and entire copy of this Petition for *Inter Partes* Review and Exhibits were provided via Federal Express to the Patent Owner by serving the correspondence address of record for the '529 Patent:

M&B IP Analysts, LLC
150 Morristown Road, Suite 205
Bernardsville, NJ 07924-2626

Further, a courtesy copy of this Petition for *Inter Partes* Review was sent via email to Patent Owner's litigation counsel:

Frederick L. Cottrell, III (cottrell@rlf.com)
Kelly E. Farnan (farnan@rlf.com)
Christine Dealy Haynes (haynes@rlf.com)
Alex Miller (alex.miller@wsgr.com)
Callie Davidson (ccdavidson@wsgr.com)
Jordan R. Jaffe (jjaffe@wsgr.com)
Lisa D. Zang (lzang@wsgr.com)
Praatika Prasad (pprasad@wsgr.com)

Respectfully submitted,

ERISE IP, P.A.

BY: /s/ Robin A. Snader
Robin A. Snader, Reg. No. 66,085
robin.snader@eriseip.com
7015 College Blvd., Suite 700
Overland Park, KS 66211